Khai phá mẫu liên hệ

Khai phá mẫu liên hệ

- Bài toán khai phá mẫu liên hệ cổ điển được định nghĩa với dữ liệu siêu thị chứa các tập hạng mục mà khách hàng mua (được gọi là giao dịch).
- Mục tiêu của bài toán là xác định các liên hệ giữa các nhóm hạng mục được mua bởi khách hàng.
- Bài toán khai phá mẫu liên hệ có nhiều ứng dụng như:
  - Dữ liệu siêu thị.
  - Khai phá văn bản.
  - Tổng quát hóa các kiểu dữ liệu định hướng phụ thuộc.
  - Một số bài toán khai phá dữ liệu khác như gom cụm, phân loại, phân tích ngoại lai.

Mô hình khai phá mẫu thường xuyên

- Bài toán khai phá mẫu thường được đặt trên dữ liệu tập hợp không thứ tự.
- Giả sử ta có cơ sở dữ liệu T với n giao dịch.
- Mỗi giao dịch được lấy từ không gian các hạng mục U.
- Các thuộc tính của mỗi giao dịch được biểu diễn dạng nhị phân.

Mô hình khai phá mẫu thường xuyên

- Ta có thí dụ sau.

Table 4.1: Example of a snapshot of a market basket data set

tid	Set of items	Binary representation
1	$\{Bread, Butter, Milk\}$	110010
2	$\{Eggs, Milk, Yogurt\}$	000111
3	$\{Bread, Cheese, Eggs, Milk\}$	101110
4	$\{Eggs, Milk, Yogurt\}$	000111
5	$\{Cheese, Milk, Yogurt\}$	001011

Source: Data Mining, Charu C. Aggarwal

Mô hình khai phá mẫu thường xuyên

Ta có các định nghĩa sau.

- Support của một tập hạng mục.

Definition 4.2.1 (Support) *The support of an itemset I is defined as the fraction of the transactions in the database $\mathcal{T} = \{T_1 \ldots T_n\}$ that contain I as a subset.*

- Các bài toán khai phá mẫu thường xuyên thường có mục tiêu xác định các tập hạng mục thỏa support cực tiểu theo yêu cầu.

Source: Data Mining, Charu C. Aggarwal

Mô hình khai phá mẫu thường xuyên

Ta có các định nghĩa sau.

- Khai phá tập hạng mục thường xuyên (định nghĩa theo biểu diễn nhị phân của giao dịch).

Definition 4.2.2 (Frequent Itemset Mining) *Given a set of transactions* $\mathcal{T} = \{T_1 \ldots T_n\}$, *where each transaction* T_i *is a subset of items from* U, *determine all itemsets* I *that occur as a subset of at least a predefined fraction minsup of the transactions in* $\mathcal{T}$.

- Ở đây support tối thiểu có dạng tỉ lệ, tuy nhiên, chúng ta cũng có thể đặt bài toán với support tối thiểu dạng số nguyên dương.

Source: Data Mining, Charu C. Aggarwal

Mô hình khai phá mẫu thường xuyên

Ta có các định nghĩa sau.

- Khai phá tập hạng mục thường xuyên (định nghĩa theo tập).

Definition 4.2.3 (Frequent Itemset Mining: Set-wise Definition) *Given a set of sets $\mathcal{T} = \{T_1 \ldots T_n\}$, where each element of the set T_i is drawn on the universe of elements U, determine all sets I that occur as a subset of at least a predefined fraction minsup of the sets in $\mathcal{T}$.*

- Ở đây support tối thiểu có dạng tỉ lệ, tuy nhiên, chúng ta cũng có thể đặt bài toán với support tối thiểu dạng số nguyên dương.

Source: Data Mining, Charu C. Aggarwal

Mô hình khai phá mẫu thường xuyên

Ta có các tính chất sau.

Property 4.2.1 (Support Monotonicity Property) *The support of every subset J of I is at least equal to that of the support of itemset I.*

$$sup(J) \geq sup(I) \quad \forall J \subseteq I \tag{4.1}$$

The monotonicity property of support implies that every subset of a frequent itemset will also be frequent. This is referred to as the *downward closure property*.

Source: Data Mining, Charu C. Aggarwal

Mô hình khai phá mẫu thường xuyên

Ta có các tính chất sau.

Property 4.2.2 (Downward Closure Property) *Every subset of a frequent itemset is also frequent.*

The downward closure property of frequent patterns is algorithmically very convenient because it provides an important constraint on the inherent structure of frequent patterns. This constraint is often leveraged by frequent pattern mining algorithms to prune the search process and achieve greater efficiency. Furthermore, the downward closure property can be used to create concise representations of frequent patterns, wherein only the *maximal* frequent subsets are retained.

Source: Data Mining, Charu C. Aggarwal

Mô hình khai phá mẫu thường xuyên

Definition 4.2.4 (Maximal Frequent Itemsets) *A frequent itemset is maximal at a given minimum support level minsup, if it is frequent, and no superset of it is frequent.*

Trong thí dụ sau, tập hạng mục {Eggs, Milk, Yogurt} là một tập hạng mục thường xuyên cực đại với support cực tiểu 0.3.

Table 4.1: Example of a snapshot of a market basket data set

tid	Set of items	Binary representation
1	$\{Bread, Butter, Milk\}$	110010
2	$\{Eggs, Milk, Yogurt\}$	000111
3	$\{Bread, Cheese, Eggs, Milk\}$	101110
4	$\{Eggs, Milk, Yogurt\}$	000111
5	$\{Cheese, Milk, Yogurt\}$	001011

Source: Data Mining, Charu C. Aggarwal

Mô hình khai phá mẫu thường xuyên

- Một tính chất thú vị của các tập hạng mục là có thể được mô tả bằng một lưới tập hạng mục.

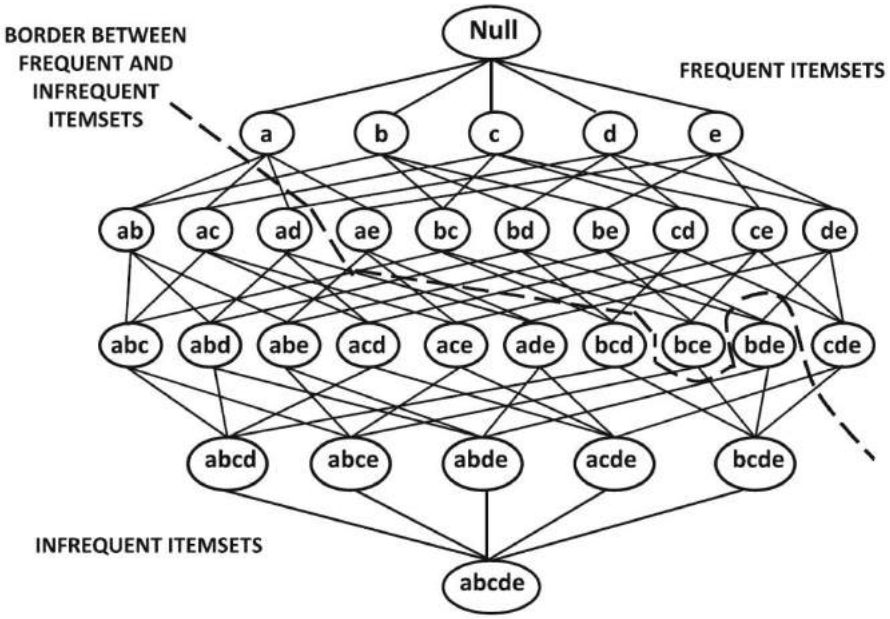


Figure 4.1: The itemset lattice

Source: Data Mining, Charu C. Aggarwal

Khuôn khổ sinh luật liên hệ

- Các tập hạng mục có thể được dùng để sinh các luật liên hệ với độ đo "độ tin cậy".

Definition 4.3.1 (Confidence) *Let X and Y be two sets of items. The confidence $conf(X \cup Y)$ of the rule $X \cup Y$ is the conditional probability of $X \cup Y$ occurring in a transaction, given that the transaction contains X. Therefore, the confidence $conf(X \Rightarrow Y)$ is defined as follows:*

$$conf(X \Rightarrow Y) = \frac{sup(X \cup Y)}{sup(X)}. \quad (4.2)$$

Source: Data Mining, Charu C. Aggarwal

Khuôn khổ sinh luật liên hệ

- Trong thí dụ này thì support của {Eggs, Milk} là 0.6, support của {Eggs, Milk, Yogurt} là 0.4.

- Từ đó, độ tin cậy của luật {Eggs, Milk} => {Eggs, Milk, Yogurt} là 0.4/0.6 = 2/3

Table 4.1: Example of a snapshot of a market basket data set

tid	Set of items	Binary representation
1	$\{Bread, Butter, Milk\}$	110010
2	$\{Eggs, Milk, Yogurt\}$	000111
3	$\{Bread, Cheese, Eggs, Milk\}$	101110
4	$\{Eggs, Milk, Yogurt\}$	000111
5	$\{Cheese, Milk, Yogurt\}$	001011

Source: Data Mining, Charu C. Aggarwal

Khuôn khổ sinh luật liên hệ

- Các luật liên hệ được định nghĩa với các tiêu chí về support và độ tin cậy.

Definition 4.3.2 (Association Rules) *Let X and Y be two sets of items. Then, the rule $X \Rightarrow Y$ is said to be an association rule at a minimum support of minsup and minimum confidence of minconf, if it satisfies both the following criteria:*

1. *The support of the itemset $X \cup Y$ is at least minsup.*
2. *The confidence of the rule $X \Rightarrow Y$ is at least minconf.*

The first criterion ensures that a sufficient number of transactions are relevant to the rule; therefore, it has the required critical mass for it to be considered relevant to the application at hand. The second criterion ensures that the rule has sufficient strength in terms of conditional probabilities. Thus, the two measures quantify different aspects of the association rule.

Source: Data Mining, Charu C. Aggarwal

Khuôn khổ sinh luật liên hệ

Khuôn khổ sinh luật quan hệ có 2 giai đoạn tương ứng với 2 tiêu chí với các ràng buộc về support và độ tin cậy.

- Trong giai đoạn đầu, tất cả tập hạng mục thường xuyên được sinh với support cực tiểu *minsup*.
- Trong giai đoạn 2, các luật liên hệ được sinh từ các tập hạng mục thường xuyên với độ tin cậy cực tiểu *minconf*.

Khuôn khổ sinh luật liên hệ

Các luật liên hệ cũng thỏa tính chất độ tin cậy đơn điệu.

Property 4.3.1 (Confidence Monotonicity) *Let X_1, X_2, and I be itemsets such that $X_1 \subset X_2 \subset I$. Then the confidence of $X_2 \Rightarrow I - X_2$ is at least that of $X_1 \Rightarrow I - X_1$.*

$$conf(X_2 \Rightarrow I - X_2) \geq conf(X_1 \Rightarrow I - X_1) \quad (4.3)$$

Source: Data Mining, Charu C. Aggarwal

Thuật toán brute force

- Với một không gian hạng mục U lớn thì việc vét cạn tất cả tập hạng mục tiềm năng cho bài toán và tính support với cơ sở dữ liệu rất không khả thi.
- Tuy nhiên, vẫn có thể giúp rút gọn thuật toán vét cạn với tính chất là không có (k+1)-mẫu (mẫu có k+1 phần tử) thường xuyên nếu không có k-mẫu thường xuyên nào (suy ra từ “downward closure property”).
- Từ đó, ta có thể đánh số và tính support của các mẫu với kích thước tăng dần (các mẫu 1 hạng mục → 2 hạng mục → … → *l* hạng mục) với
- Tức là tính support các mẫu 1 hạng mục → 2 hạng mục → … → *l* hạng mục với không *l*-mẫu nào là thường xuyên và kết thúc thuật toán tại đó.

Thuật toán brute force

- Chúng ta có các cách tiếp cận sau để tăng hiệu suất thuật toán
  - Giảm kích thước không gian tìm kiếm bằng cách cắt giảm các tập hạng mục tiềm năng.
  - Tính support của mỗi tập hạng mục tiềm năng hiệu quả hơn bằng cách cắt giảm các giao dịch không quan trọng.
  - Sử dụng các cấu trúc dữ liệu gọn gàng để biểu diễn các dữ liệu cần thiết khi tính support.

Thuật toán Apriori

- Tính chất “downward closure” áp đặt một cấu trúc rõ ràng cho tập các mẫu thường xuyên.
- Thuật toán Apriori sử dụng tính chất “downward closure” để cắt giảm không gian tìm kiếm của bài toán.
  - Thông tin về sự không thường xuyên của các tập hạng mục có thể được dùng để sinh các tập mẹ tiềm năng
  - Từ đó, nếu một tập hạng mục không thường xuyên thì không xét các tập mẹ của tập đó.

Thuật toán Apriori

Algorithm *Apriori*(Transactions: $\mathcal{T}$, Minimum Support: *minsup*)
begin
  $k = 1$;
  $\mathcal{F}_1 = \{$ All Frequent 1-itemsets $\}$;
  while $\mathcal{F}_k$ is not empty **do begin**
    Generate $\mathcal{C}_{k+1}$ by joining itemset-pairs in $\mathcal{F}_k$;
    Prune itemsets from $\mathcal{C}_{k+1}$ that violate downward closure;
    Determine $\mathcal{F}_{k+1}$ by support counting on $(\mathcal{C}_{k+1}, \mathcal{T})$ and retaining
      itemsets from $\mathcal{C}_{k+1}$ with support at least *minsup*;
    $k = k + 1$;
  end;
  return$(\cup_{i=1}^{k} \mathcal{F}_i)$;
end

Figure 4.2: The *Apriori* algorithm

Source: Data Mining, Charu C. Aggarwal

Thuật toán Apriori

- Để tính support một cách hiệu quả, thuật toán Apriori sử dụng cấu trúc dữ liệu hash tree để xác định mỗi tập hạng mục tiềm năng có nằm trong một giao dịch hay không.

Thuật toán cây đánh số

- Cây đánh số này cũng là một subgraph của lưới tập hạng mục.
- Với các thuật toán này, các tập hạng mục tiềm năng được sinh trong một cấu trúc cây đánh số.
- Cấu trúc này còn được gọi là cây từ điển do nó phụ thuộc vào thứ tự từ điển giữa các hạng mục.
- Các mẫu tiềm năng được sinh bằng việc mở rộng cây này.
- Thuật toán Aprori cũng có thể được xem là một dạng cụ thể của thuật toán cây đánh số.

Thuật toán cây đánh số

- Cây đánh số này xác định trên các tập hạng mục thường xuyên như sau.

1. A node exists in the tree corresponding to each frequent itemset. The root of the tree corresponds to the *null* itemset.

2. Let $I = \{i_1, \ldots i_k\}$ be a frequent itemset, where $i_1, i_2 \ldots i_k$ are listed in lexicographic order. The parent of the node I is the itemset $\{i_1, \ldots i_{k-1}\}$. Thus, the child of a node can only be extended with items occurring lexicographically *after* all items occurring in that node. The enumeration tree can also be viewed as a *prefix* tree on the lexicographically ordered string representation of the itemsets.

Source: Data Mining, Charu C. Aggarwal

Thuật toán cây đánh số

- Cây đánh số này xác định trên các tập hạng mục thường xuyên như sau.

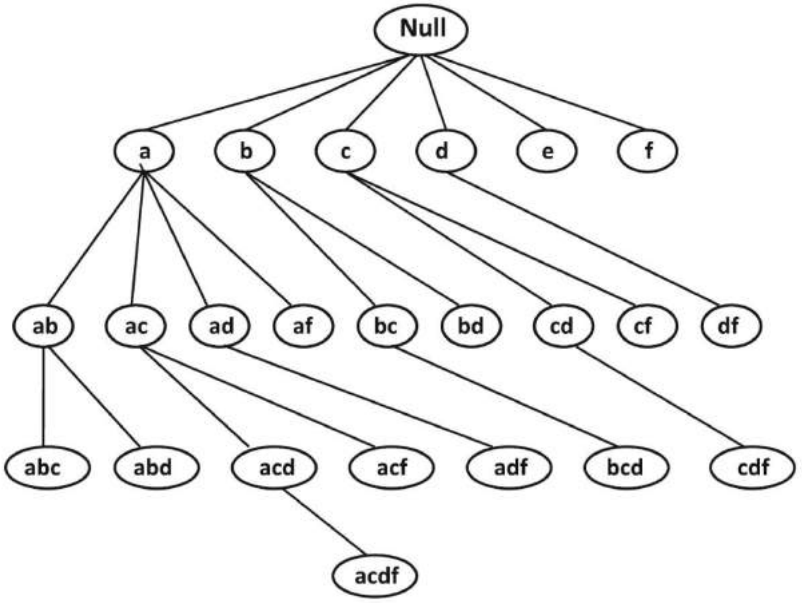


Figure 4.3: The lexicographic or enumeration tree of frequent itemsets

Source: Data Mining, Charu C. Aggarwal

Các thuật toán khai phá tập hạng mục thường xuyên

Thuật toán cây đánh số

(TreeProjection và DepthProject)

- TreeProjection là một họ các phương pháp sử dụng các phép chiếu đệ quy từ các giao dịch xuống cấu trúc cây đánh số.
- Mục đích của các phép chiếu đệ quy này là để tái sử dụng phần tính support đã được làm sẵn tại mỗi node trong cây đánh số, giúp giảm khối lượng tính toán rất nhiều.
- DepthProject là một trong các cách tiếp cận cụ thể của TreeProjection.

Các thuật toán khai phá tập hạng mục thường xuyên

Thuật toán cây đánh số

(TreeProjection và DepthProject)

Algorithm *ProjectedEnumerationTree*(Transactions: $\mathcal{T}$,
Minimum Support: $minsup$)
begin
  Initialize enumeration tree $\mathcal{ET}$ to a single $(Null, \mathcal{T})$ root node;
  while any node in $\mathcal{ET}$ has not been examined **do begin**
    Select an unexamined node $(P, \mathcal{T}(P))$ from $\mathcal{ET}$ for examination;
    Generate candidates item extensions $C(P)$ of node $(P, \mathcal{T}(P))$;
    Determine frequent item extensions $F(P) \subseteq C(P)$ by support counting
      of individual items in smaller projected database $\mathcal{T}(P)$;
    Remove infrequent items in $\mathcal{T}(P)$;
    for each frequent item extension $i \in F(P)$ **do begin**
      Generate $\mathcal{T}(P \cup \{i\})$ from $\mathcal{T}(P)$;
      Add $(P \cup \{i\}, \mathcal{T}(P \cup \{i\}))$ as child of P in $\mathcal{ET}$;
    end
  end
  return enumeration tree $\mathcal{ET}$;
end

Figure 4.5: Generic enumeration-tree growth with unspecified growth strategy and database projections

Source: Data Mining, Charu C. Aggarwal

Thuật toán cây đánh số

(Phương pháp đếm dọc)

- Cơ sở dữ liệu giao dịch T cũng có thể được biểu diễn với cách biểu diễn cơ sở dữ liệu dọc.

Table 4.2: Vertical representation of market basket data set

Item	Set of tids	Binary representation
Bread	$\{1, 3\}$	10100
Butter	$\{1\}$	10000
Cheese	$\{3, 5\}$	00101
Eggs	$\{2, 3, 4\}$	01110
Milk	$\{1, 2, 3, 4, 5\}$	11111
Yogurt	$\{2, 4, 5\}$	01011

Source: Data Mining, Charu C. Aggarwal

Thuật toán cây đánh số

(Phương pháp đếm dọc)

- Với cách biểu diễn này, mỗi hạng mục được gắn với một danh sách các id xác định giao dịch.

Table 4.2: Vertical representation of market basket data set

Item	Set of tids	Binary representation
Bread	$\{1, 3\}$	10100
Butter	$\{1\}$	10000
Cheese	$\{3, 5\}$	00101
Eggs	$\{2, 3, 4\}$	01110
Milk	$\{1, 2, 3, 4, 5\}$	11111
Yogurt	$\{2, 4, 5\}$	01011

Source: Data Mining, Charu C. Aggarwal

Các thuật toán khai phá tập hạng mục thường xuyên

Thuật toán cây đánh số

(Phương pháp đếm dọc)

- Cách biểu diễn này cũng có thể được xem là sử dụng chuyển vị của ma trận nhị phân biểu diễn các giao dịch.

Table 4.1: Example of a snapshot of a market basket data set

tid	Set of items	Binary representation
1	$\{Bread, Butter, Milk\}$	110010
2	$\{Eggs, Milk, Yogurt\}$	000111
3	$\{Bread, Cheese, Eggs, Milk\}$	101110
4	$\{Eggs, Milk, Yogurt\}$	000111
5	$\{Cheese, Milk, Yogurt\}$	001011

Table 4.2: Vertical representation of market basket data set

Item	Set of tids	Binary representation
$Bread$	$\{1, 3\}$	10100
$Butter$	$\{1\}$	10000
$Cheese$	$\{3, 5\}$	00101
$Eggs$	$\{2, 3, 4\}$	01110
$Milk$	$\{1, 2, 3, 4, 5\}$	11111
$Yogurt$	$\{2, 4, 5\}$	01011

Source: Data Mining, Charu C. Aggarwal

Thuật toán cây đánh số

(Phương pháp đếm dọc)

- Từ cách biểu diễn này, chúng ta cũng có thuật toán Apriori dọc.

Algorithm *VerticalApriori*(Transactions: $\mathcal{T}$, Minimum Support: *minsup*)
begin
  $k = 1$;
  $\mathcal{F}_1 = \{$ All Frequent 1-itemsets $\}$;
  Construct vertical *tid* lists of each frequent item;
  while $\mathcal{F}_k$ is not empty **do begin**
    Generate $\mathcal{C}_{k+1}$ by joining itemset-pairs in $\mathcal{F}_k$;
    Prune itemsets from $\mathcal{C}_{k+1}$ that violate downward closure;
    Generate *tid* list of each candidate itemset in $\mathcal{C}_{k+1}$ by intersecting
      tid lists of the itemset-pair in $\mathcal{F}_k$ that was used to create it;
    Determine supports of itemsets in $\mathcal{C}_{k+1}$ using lengths of their *tid* lists;
    $\mathcal{F}_{k+1}$= Frequent itemsets of $\mathcal{C}_{k+1}$ together with their *tid* lists;
    $k = k + 1$;
  end;
  return$(\cup_{i=1}^{k} \mathcal{F}_i)$;
end

Figure 4.7: The vertical *Apriori* algorithm of Savasere et al. [446]

Source: Data Mining, Charu C. Aggarwal

Khai phá mẫu liên hệ (tiếp theo)

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Các cây đánh số được xây dựng bằng cách mở rộng các tiền tố (prefix) của các tập hạng mục với thứ tự từ điển.
- Tuy nhiên, chúng ta cũng có thể biểu diễn một số lớp phương pháp khám phá tập danh mục theo cách đệ quy với khám phá theo hậu tố (suffix).

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Các phương pháp phát triển mẫu theo hậu tố đệ quy thường được hiểu với cấu trúc dữ liệu cây mẫu thường xuyên (FP-tree).
- Cấu trúc cây mẫu thường xuyên này giúp chúng ta thực hiện các thuật toán khám phá mẫu đệ quy một cách tiết kiệm.
- Tuy nhiên, các thuật toán này cũng có thể được thực hiện với mảng (array) và pointer.

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

Algorithm *RecursiveSuffixGrowth*(Transactions in terms of frequent 1-items: $\mathcal{T}$, Minimum Support: $minsup$, Current Suffix: P)
begin
  for each item i in $\mathcal{T}$ **do begin**
    report itemset $P_i = \{i\} \cup P$ as frequent;
    Extract all transactions $\mathcal{T}_i$ from $\mathcal{T}$ containing item i;
    Remove all items from $\mathcal{T}_i$ that are lexicographically $\geq i$;
    Remove all infrequent items from $\mathcal{T}_i$;
    if $(\mathcal{T}_i \neq \phi)$ **then** *RecursiveSuffixGrowth*$(\mathcal{T}_i, minsup, P_i)$;
  end
end

Figure 4.8: Generic recursive suffix growth on transaction database expressed in terms of frequent 1-items

Source: Data Mining, Charu C. Aggarwal

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

Thực hiện với mảng nhưng không có pointer

- Các tập giao dịch và giao dịch điều kiện trong thuật toán có thể được biểu diễn bằng các mảng.
- Các vòng lặp for khi sử dụng mảng rất **tốn kém tính toán**.
- Để có trade off tốt hơn giữa yêu cầu **tính toán** và yêu cầu **lưu trữ**, chúng ta có thể sử dụng thêm pointer (cần nhiều lưu trữ hơn nhưng ít tính toán hơn).

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Thực hiện với pointer nhưng không có cây mẫu thường xuyên

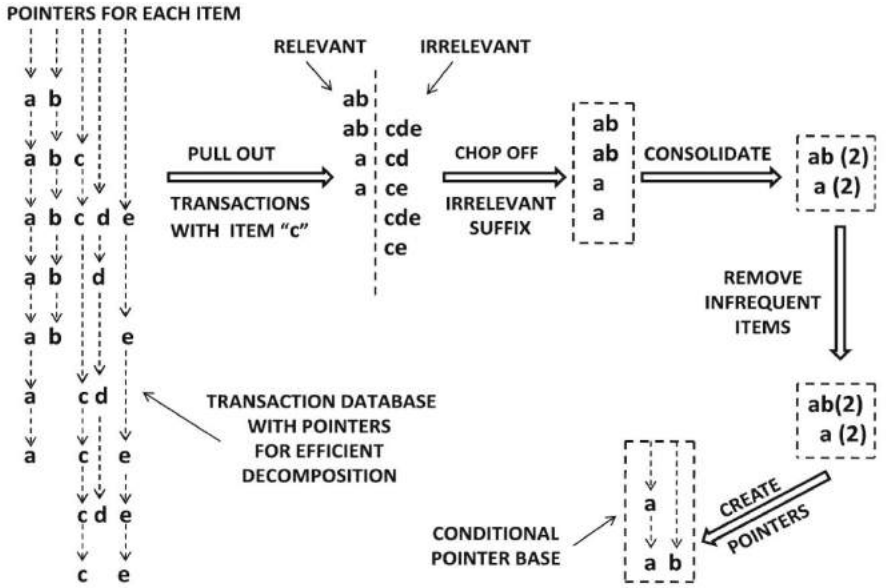


Figure 4.9: Illustration of recursive pattern growth with pointers and no FP-Tree

Source: Data Mining, Charu C. Aggarwal

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Thực hiện với pointer nhưng không có cây mẫu thường xuyên

Algorithm *RecursiveGrowthPointers*(Transactions in terms of frequent 1-items: $\mathcal{T}$, Minimum Support: *minsup*, Current Suffix: P)
begin
  for each item i in $\mathcal{T}$ **do begin**
    report itemset $P_i = \{i\} \cup P$ as frequent;
    Use pointers to extract all transactions $\mathcal{T}_i$ from $\mathcal{T}$ containing item i;
    Remove all items from $\mathcal{T}_i$ that are lexicographically $\geq i$;
    Remove all infrequent items from $\mathcal{T}_i$;
    Set up pointers for $\mathcal{T}_i$;
    if $(\mathcal{T}_i \neq \phi)$ **then** *RecursiveGrowthPointers*$(\mathcal{T}_i, minsup, P_i)$;
  end
end

Figure 4.10: Generic recursive suffix growth with pointers

Source: Data Mining, Charu C. Aggarwal

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Thực hiện với pointer và cây mẫu thường xuyên
  - Cấu trúc cây mẫu thường xuyên (FP-tree) được thiết kế với mục đích hiệu quả về **lưu trữ**.
  - Cấu trúc này có thể dùng để biểu diễn cơ sở dữ liệu các giao dịch thay cho các mảng.
  - Ở thực hiện này của thuật toán, mảng được thay bằng các FP-tree nhưng các pointer vẫn được sử dụng.

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Thực hiện với pointer nhưng không có cây mẫu thường xuyên

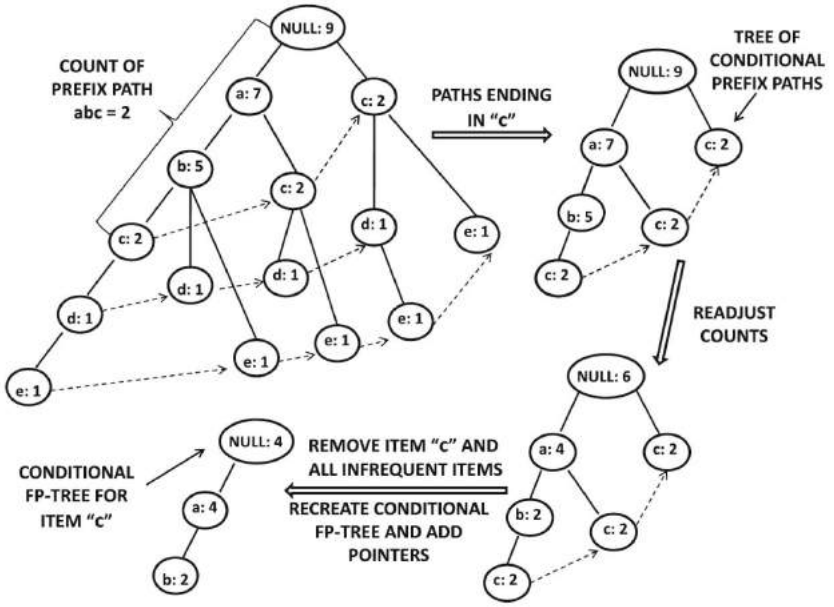


Figure 4.11: Illustration of recursive pattern growth with pointers and FP-Tree

Source: Data Mining, Charu C. Aggarwal

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Thực hiện với pointer nhưng không có cây mẫu thường xuyên

Algorithm *FP-growth*(FP-Tree of frequent items: $\mathcal{FPT}$, Minimum Support: *minsup*, Current Suffix: P)
begin
  if $\mathcal{FPT}$ is a single path
    then determine all combinations C of nodes on the path, and report $C \cup P$ as frequent;
  else (Case when $\mathcal{FPT}$ is not a single path)
  for each item i in $\mathcal{FPT}$ **do begin**
    report itemset $P_i = \{i\} \cup P$ as frequent;
    Use pointers to extract conditional prefix paths from $\mathcal{FPT}$ containing item i;
    Readjust counts of prefix paths and remove i;
    Remove infrequent items from prefix paths and reconstruct conditional FP-Tree $\mathcal{FPT}_i$;
    if $(\mathcal{FPT}_i \neq \phi)$ **then** *FP-growth*$(\mathcal{FPT}_i, minsup, P_i)$;
  end
end

Figure 4.12: The *FP-growth* algorithm with an FP-Tree representation of the transaction database expressed in terms of frequent 1-items

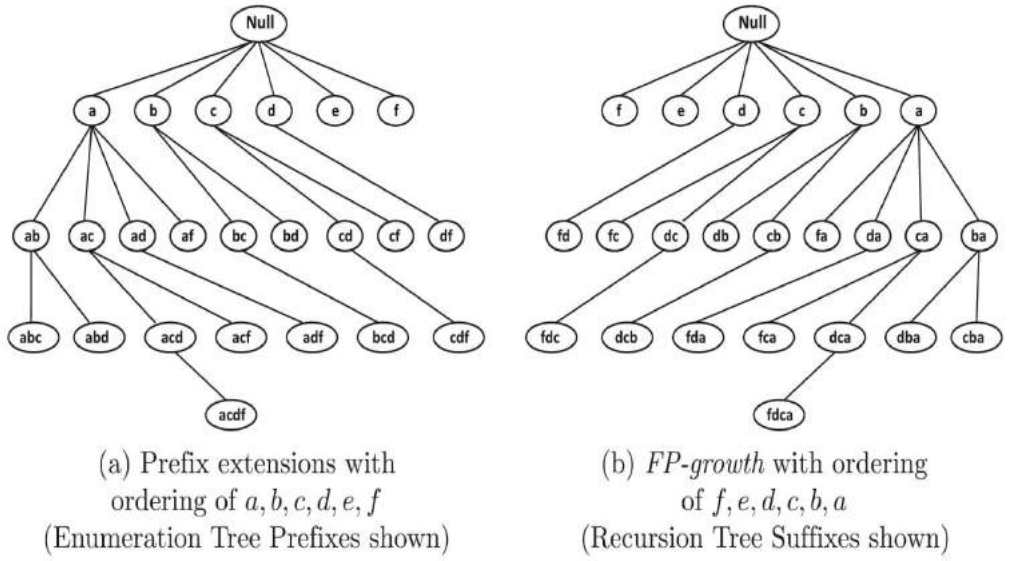
Source: Data Mining, Charu C. Aggarwal

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Trade-off giữa các cấu trúc dữ liệu khác nhau
  - Về yêu cầu lưu trữ, thực hiện thuật toán với FP-tree nhẹ hơn thực hiện với pointer.
  - Tuy nhiên, thực hiện với FP-tree cũng có thể nặng lưu trữ hơn thực hiện với mảng.
  - Vấn đề hạn chế trong thực tế thường là việc bộ nhớ có chứa đủ cơ sở dữ liệu giao dịch hay không.

Các phương pháp phát triển mẫu dựa vào hậu tố đệ quy

- Quan hệ giữa phương pháp phát triển mẫu thường xuyên và cây đánh số
- Phương pháp phát triển cây mẫu thường xuyên có sự tương ứng với phương pháp cây đánh số.



(a) Prefix extensions with ordering of a, b, c, d, e, f (Enumeration Tree Prefixes shown)

(b) *FP-growth* with ordering of f, e, d, c, b, a (Recursion Tree Suffixes shown)

Figure 4.13: Enumeration trees are identical to *FP-growth* recursion trees with reverse lexicographic ordering

Source: Data Mining, Charu C. Aggarwal

Các mô hình khác

- Mô hình truyền thống cho việc tìm tập hạng mục thường xuyên được sử dụng rộng rãi do tính đơn giản.
- Sự đơn giản quan trọng này nằm ở.
  - Sử dụng tần suất đơn thuần cho khi tính support các tập hạng mục.
  - Sử dụng xác suất điều kiện để tính độ tin cậy.
- Trong một số ứng dụng, chúng ta muốn có các độ đo khác thích hợp hơn.

Các phương pháp khác

- Hệ số thống kê cho sự tương quan

A natural statistical measure is the Pearson coefficient of correlation between a pair of items. The Pearson coefficient of correlation between a pair of random variables X and Y is defined as follows:

$$\rho = \frac{E[X \cdot Y] - E[X] \cdot E[Y]}{\sigma(X) \cdot \sigma(Y)}. \tag{4.4}$$

In the case of market basket data, X and Y are binary variables whose values reflect presence or absence of items. The notation $E[X]$ denotes the expectation of X, and $\sigma(X)$ denotes the standard deviation of X. Then, if $sup(i)$ and $sup(j)$ are the relative supports of individual items, and $sup(\{i, j\}$ is the relative support of itemset $\{i, j\}$, then the overall correlation can be estimated from the data as follows:

$$\rho_{ij} = \frac{sup(\{i, j\}) - sup(i) \cdot sup(j)}{\sqrt{sup(i) \cdot sup(j) \cdot (1 - sup(i)) \cdot (1 - sup(j))}}. \tag{4.5}$$

The coefficient of correlation always lies in the range $[-1, 1]$, where the value of $+1$ indicates perfect positive correlation, and the value of -1 indicates perfect negative correlation. A value near 0 indicates weakly correlated data. This measure satisfies the bit symmetric property. While the coefficient of correlation is statistically considered the most robust way of measuring correlations, it is often intuitively hard to interpret when dealing with items of varying but low support values.

Source: Data Mining, Charu C. Aggarwal

Các phương pháp khác

- Độ đo χ^2 là một độ đo khác xem sự có mặt và vắng mặt của các hạng mục một cách tương tự.
- Cho *X* là một tập gồm k biến ngẫu nhiên nhị phân (đại diện cho các hạng mục).
- Với O_i *và* E_i lần lượt là giá trị quan sát được và giá trị kì vọng của support tuyệt đối của trạng thái *i*

$$\chi^2(X) = \sum_{i=1}^{2^{|X|}} \frac{(O_i - E_i)^2}{E_i}.$$

Source: Data Mining, Charu C. Aggarwal

Các phương pháp khác

- Tỉ lệ interest

The interest ratio is a simple and intuitively interpretable measure. The interest ratio of a set of items $\{i_1 \ldots i_k\}$ is denoted as $I(\{i_1, \ldots i_k\})$, and is defined as follows:

$$I(\{i_1 \ldots i_k\}) = \frac{sup(\{i_1 \ldots i_k\})}{\prod_{j=1}^{k} sup(i_j)}. \tag{4.7}$$

When the items are statistically independent, the joint support in the numerator will be equal to the product of the supports in the denominator. Therefore, an interest ratio of 1 is the break-even point. A value greater than 1 indicates that the variables are positively correlated, whereas a ratio of less than 1 is indicative of negative correlation.

When some items are extremely rare, the interest ratio can be misleading. For example, if an item occurs in only a single transaction in a large transaction database, each item that co-occurs with it in that transaction can be paired with it to create a 2-itemset with a very high interest ratio. This is statistically misleading. Furthermore, because the interest ratio does not satisfy the downward closure property, it is difficult to design efficient algorithms for computing it.

Source: Data Mining, Charu C. Aggarwal

Các phương pháp khác

Độ do sự tin cậy đối xứng

- Độ đo sự tin cậy truyền thống không đối xứng giữa tiền kiện và hệ quả, trong khi độ đo support đối xứng.
- Chúng ta có thể thay độ đó sự tin cậy này bằng một độ đo đối xứng.

Các phương pháp khác

- Hệ số cosine trên các cột

The cosine coefficient is usually applied to the rows to determine the similarity among transactions. However, it can also be applied to the columns, to determine the similarity between items. The cosine coefficient is best computed using the vertical *tid* list representation on the corresponding binary vectors. The cosine value on the binary vectors computes to the following:

$$cosine(i, j) = \frac{sup(\{i, j\})}{\sqrt{sup(i)} \cdot \sqrt{sup(j)}}. \quad (4.8)$$

The numerator can be evaluated as the length of the intersection of the *tid* lists of items i and j. The cosine measure can be viewed as the geometric mean of the confidences of the rules $\{i\} \Rightarrow \{j\}$ and $\{j\} \Rightarrow \{i\}$. Therefore, the cosine is a kind of symmetric confidence measure.

Source: Data Mining, Charu C. Aggarwal

Các phương pháp khác

- Hệ số Jaccard và kĩ thuật Min-hash

The Jaccard coefficient was introduced in Chap. 3 to measure similarity between sets. The *tid* lists on a column can be viewed as a set, and the Jaccard coefficient between two *tid* lists can be used to compute the similarity. Let S_1 and S_2 be two sets. As discussed in Chap. 3, the Jaccard coefficient $J(S_1, S_2)$ between the two sets can be computed as follows:

$$J(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}. \tag{4.9}$$

The Jaccard coefficient can easily be generalized to multiway sets, as follows:

$$J(S_1 \ldots S_k) = \frac{|\cap S_i|}{|\cup S_i|}. \tag{4.10}$$

Source: Data Mining, Charu C. Aggarwal

Các phương pháp khác

- Collective strength

The collective strength of an itemset is defined in terms of its *violation rate*. An itemset I is said to be in *violation* of a transaction, if some of the items are present in the transaction, and others are not. The *violation rate* $v(I)$ of an itemset I is the fraction of violations of the itemset I over all transactions. The *collective strength* $C(I)$ of an itemset I is defined in terms of the violation rate as follows:

$$C(I) = \frac{1 - v(I)}{1 - E[v(I)]} \cdot \frac{E[v(I)]}{v(I)}. \tag{4.11}$$

Source: Data Mining, Charu C. Aggarwal

Các phương pháp khác

- Collective strength

Intuitively, if the violation of an itemset in a transaction is a "bad event" from the perspective of trying to establish a high correlation among items, then $v(I)$ is the fraction of bad events, and $(1 - v(I))$ is the fraction of "good events." Therefore, collective strength may be understood as follows:

$$C(I) = \frac{\text{Good Events}}{\text{E[Good Events]}} \cdot \frac{\text{E[Bad Events]}}{\text{Bad Events}}. \quad (4.13)$$

Source: Data Mining, Charu C. Aggarwal

Các phương pháp khác

- Collective strength

The concept of collective-strength may be strengthened to *strongly collective* itemsets.

Definition 4.5.1 *An itemset I is denoted to be strongly collective at level s, if it satisfies the following properties:*

1. *The collective strength $C(I)$ of the itemset I is at least s.*
2. **Closure property:** *The collective strength $C(J)$ of every subset J of I is at least s.*

Source: Data Mining, Charu C. Aggarwal

Các phương pháp khác

Quan hệ với khai phá mẫu negative

- Trong một số ứng dụng, chúng ta muốn xác định mẫu giữa các hạng mục hoặc giữa sự vắng mặt của các hạng mục.
- Các ứng dụng khai phá như cần độ đo đối xứng để xử lý sự có mặt hay vắng mặt của các hạng mục như nhau.
- Một số độ đo như hệ số Jaccard và collective strength thỏa tính chất downward closure có thể được sử dụng cho việc này.

Một số thuật toán meta có ích

- Một thuật toán meta là một thuật toán có sử dụng một thuật toán nào đó bên trong dưới dạng một subroutine.
- Việc này có thể giúp thuật toán chính hiệu quả hơn hoặc có thêm hiểu biết mới.
- Có 2 loại thuật toán meta thường dùng trong khai phá mẫu.
  - Loại sử dụng việc lấy mẫu (sampling) để cải thiện hiệu năng.
  - Loại sử dụng các subrountine tiền xử lý hoặc hậu xử lý để áp dụng vào các trường hợp khác.

Một số thuật toán meta có ích

Các phương pháp lấy mẫu

- Khi cơ sở dữ liệu giao dịch quá lớn, không thể chứa trong bộ nhớ chính thì chúng ta có thể dùng các phương pháp lấy mẫu.
- Khi dùng thuật toán khai phá tập hạng mục với dữ liệu đươc lấy mẫu, chúng ta cần quan tâm 2 thử thách sau.
  - False positive: các mẫu thỏa trên dữ liệu được lấy mẫu nhưng không thỏa trên dữ liệu gốc.
  - False negative: các mẫu không thỏa trên dữ liệu được lấy mẫu nhưng thỏa trên dữ liệu gốc.

Một số thuật toán meta có ích

Ensemble phân hoạch dữ liệu

- Một cách tiếp cận đảm bảo đươc không có false positive và false negative là sử dụng các ensemble được phân hoạch.
- Cách này có thể được dùng để giảm tốn kém truy suất lưu trữ hoặc giảm tốn kém bộ nhớ.

Một số thuật toán meta có ích

Tổng quát hóa đến các kiểu dữ liệu khác

- Việc tổng quát hóa đến các kiểu dữ liệu khác có thể được thực hiện với các phương pháp biến đổi kiểu dữ liệu.